Network Security Lab – 4

- **Tcp and Udp Port Scanning**

➔ ***Srinivas Piskala Ganesh Babu – N13138339 and spg349***

1. ***For TCP you will scan all the ports from 0 to 100. You will collect the responses and sort them by their status of OPEN, CLOSED, FILTERED. i.e. OPEN:1, 2, 3, ...; CLOSED: 0, 8, 9, ...; FILTERED: 11, 12, ... You must account for dropped packets. If you send out a TCP/SYN packet and get no response, you should send out another as the packet may have been dropped.***

Solution Highlights:

✓ Scanned Ports 1-100

✓ Used Retry Argument in Scapy as well as ensured to retry one more time manually

✓ Compared with the NMAP Results and Confirmed it to match the findings

➔ **Results were: Open: 53, Closed: [List of all ports from 1-100 except 53]**

Program Flow:

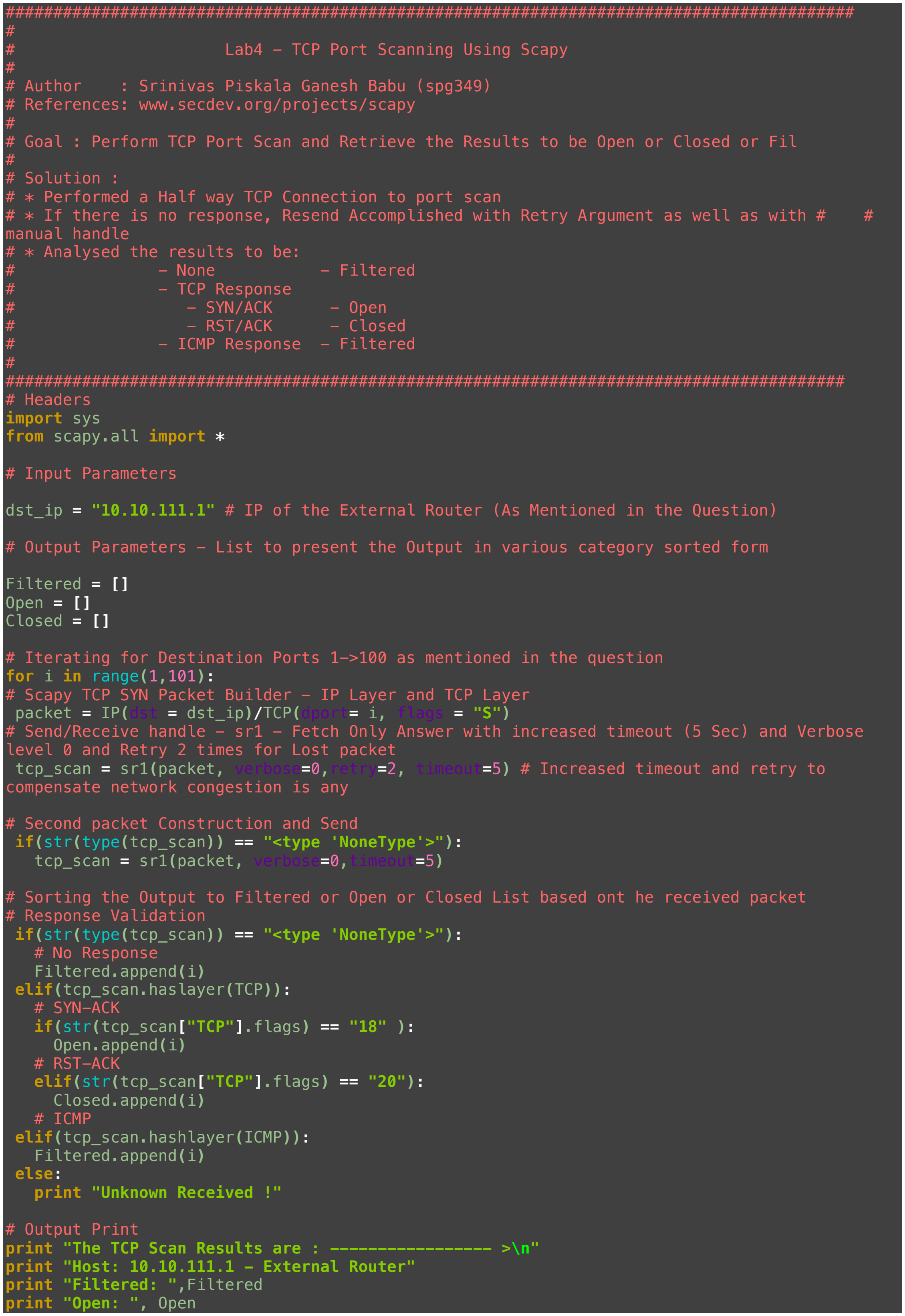
- Performed a Half way TCP Connection to port scan
- If there is no response, Resend Accomplished with Retry Argument as well as with manual handle
- Analyzed the results to be:

```
        - None                - Filtered
        - TCP Response
          - SYN/ACK           - Open
          - RST/ACK           - Closed
        - ICMP Response       - Filtered
```

Highlights:

- Used Lists for categories Open, Closed and Filtered sets
- Hardcoded IP of the External Router – 10.10.111.1
- Increased timeout and retry to compensate the network congestion
- Iterated the Destination port of the TCP setting from 1 -> 101 for scanning the 100 ports
- RST/ACK not used because of the Halfway connect scan considered

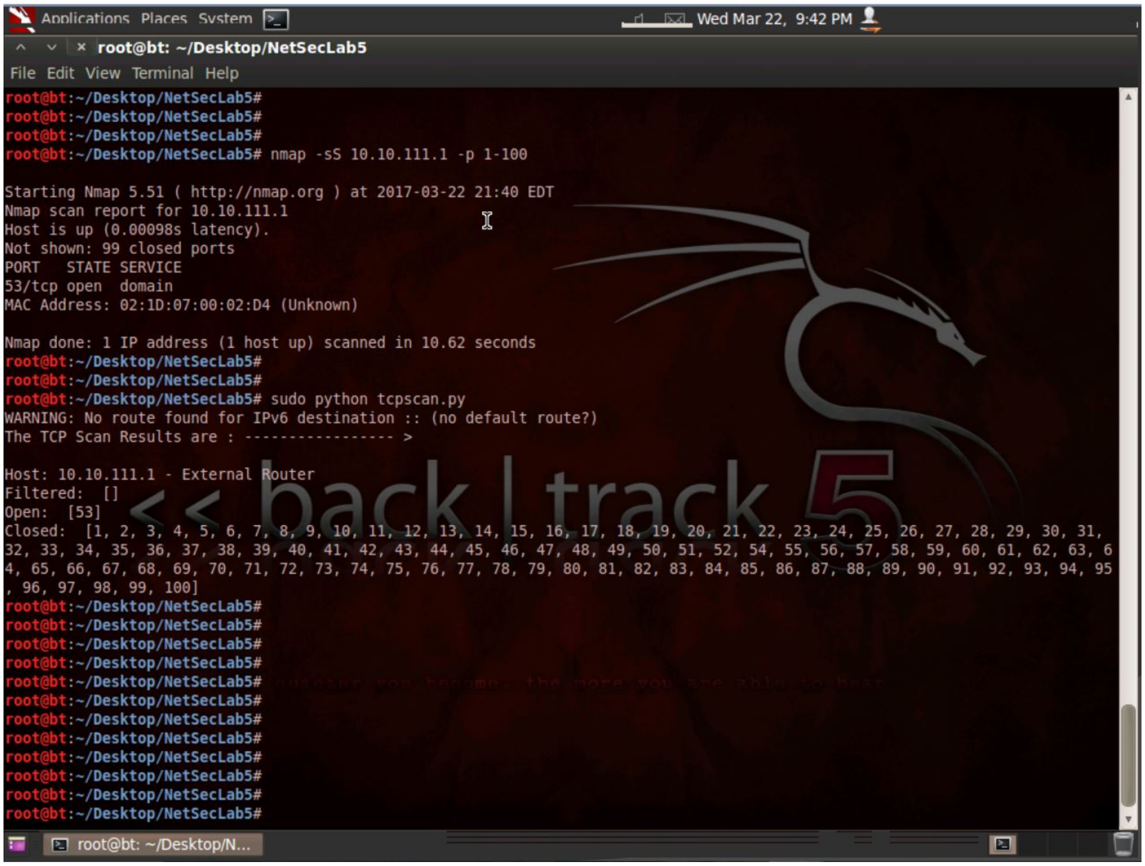
- **Code:**



```
#########################################################################################
#
#                       Lab4 - TCP Port Scanning Using Scapy
#
# Author    : Srinivas Piskala Ganesh Babu (spg349)
# References: www.secdev.org/projects/scapy
#
# Goal : Perform TCP Port Scan and Retrieve the Results to be Open or Closed or Fil
#
# Solution :
# * Performed a Half way TCP Connection to port scan
# * If there is no response, Resend Accomplished with Retry Argument as well as with #    #
manual handle
# * Analysed the results to be:
#                - None            - Filtered
#                - TCP Response
#                   - SYN/ACK       - Open
#                   - RST/ACK       - Closed
#                - ICMP Response  - Filtered
#
########################################################################################
# Headers
import sys
from scapy.all import *

# Input Parameters

dst_ip = "10.10.111.1" # IP of the External Router (As Mentioned in the Question)

# Output Parameters - List to present the Output in various category sorted form

Filtered = []
Open = []
Closed = []

# Iterating for Destination Ports 1->100 as mentioned in the question
for i in range(1,101):
# Scapy TCP SYN Packet Builder - IP Layer and TCP Layer
 packet = IP(dst = dst_ip)/TCP(dport= i, flags = "S")
# Send/Receive handle - sr1 - Fetch Only Answer with increased timeout (5 Sec) and Verbose
level 0 and Retry 2 times for Lost packet
 tcp_scan = sr1(packet, verbose=0,retry=2, timeout=5) # Increased timeout and retry to
compensate network congestion is any

# Second packet Construction and Send
 if(str(type(tcp_scan)) == "<type 'NoneType'>"):
   tcp_scan = sr1(packet, verbose=0,timeout=5)

# Sorting the Output to Filtered or Open or Closed List based ont he received packet
# Response Validation
 if(str(type(tcp_scan)) == "<type 'NoneType'>"):
   # No Response
   Filtered.append(i)
 elif(tcp_scan.haslayer(TCP)):
   # SYN-ACK
   if(str(tcp_scan["TCP"].flags) == "18" ):
     Open.append(i)
   # RST-ACK
   elif(str(tcp_scan["TCP"].flags) == "20"):
     Closed.append(i)
   # ICMP
 elif(tcp_scan.hashlayer(ICMP)):
   Filtered.append(i)
 else:
   print "Unknown Received !"

# Output Print
print "The TCP Scan Results are : ----------------- >\n"
print "Host: 10.10.111.1 - External Router"
print "Filtered: ",Filtered
print "Open: ", Open
print "Closed: ",Closed
```

Results Capture: - Screen shows program output results and nmap results

- The Screen shows the Nmap results to match the Scapy TCP Scan results


Applications Places System
Wed Mar 22, 9:42 PM
root@bt: ~/Desktop/NetSecLab5
File Edit View Terminal Help
root@bt: ~/Desktop/N...

```
root@bt:~/Desktop/NetSecLab5#
root@bt:~/Desktop/NetSecLab5#
root@bt:~/Desktop/NetSecLab5#
root@bt:~/Desktop/NetSecLab5# nmap -sS 10.10.111.1 -p 1-100

Starting Nmap 5.51 ( http://nmap.org ) at 2017-03-22 21:40 EDT
Nmap scan report for 10.10.111.1
Host is up (0.00098s latency).
Not shown: 99 closed ports
PORT   STATE SERVICE
53/tcp open  domain
MAC Address: 02:1D:07:00:02:D4 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 10.62 seconds
root@bt:~/Desktop/NetSecLab5#
root@bt:~/Desktop/NetSecLab5#
root@bt:~/Desktop/NetSecLab5# sudo python tcpscan.py
WARNING: No route found for IPv6 destination :: (no default route?)
The TCP Scan Results are : ---------------- >

Host: 10.10.111.1 - External Router
Filtered:  []
Open:  [53]
Closed:  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,
32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 6
4, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95
, 96, 97, 98, 99, 100]
root@bt:~/Desktop/NetSecLab5#
root@bt:~/Desktop/NetSecLab5#
root@bt:~/Desktop/NetSecLab5#
root@bt:~/Desktop/NetSecLab5#
root@bt:~/Desktop/NetSecLab5#
root@bt:~/Desktop/NetSecLab5#
root@bt:~/Desktop/NetSecLab5#
root@bt:~/Desktop/NetSecLab5#
root@bt:~/Desktop/NetSecLab5#
root@bt:~/Desktop/NetSecLab5#
root@bt:~/Desktop/NetSecLab5#
root@bt:~/Desktop/NetSecLab5#
```

2. ***For UDP you will also scan all the ports from 0 to 100 and collect the responses by their status of OPEN, CLOSED, or OPEN|FILTERED (open or filtered). Remember to account for packets being dropped. With UDP, no response means either the port is OPEN or the packet was FILTERED, or the packet was dropped. You should send out additional UDP packets to verify.***

Solution Highlights:

- ✓ Scanned Ports 1-100
- ✓ Used Retry Argument in Scapy to perform more than one retry when Nothing is received, performed retry manually too.
- ✓ Compared with the NMAP Results and Confirmed it to match the findings
- ➔ **Results were: Open|Filtered: 53,67,68, Closed: [List of all ports from 1-100 except 53,67,68], Open: 0**

Program Flow:

- Performed a UDP Packet Send to port scan
- If there is no response, Resend Accomplished with Retry Argument as well as with manual handle
- Analyzed the results to be:

  - None - Open|Filtered
  - UDP Response - Open
  - ICMP Response
    - Type 3 and Code 3 – Destination and Port Unreachable – Closed
    - Else consider as - Open|Filtered

Highlights:

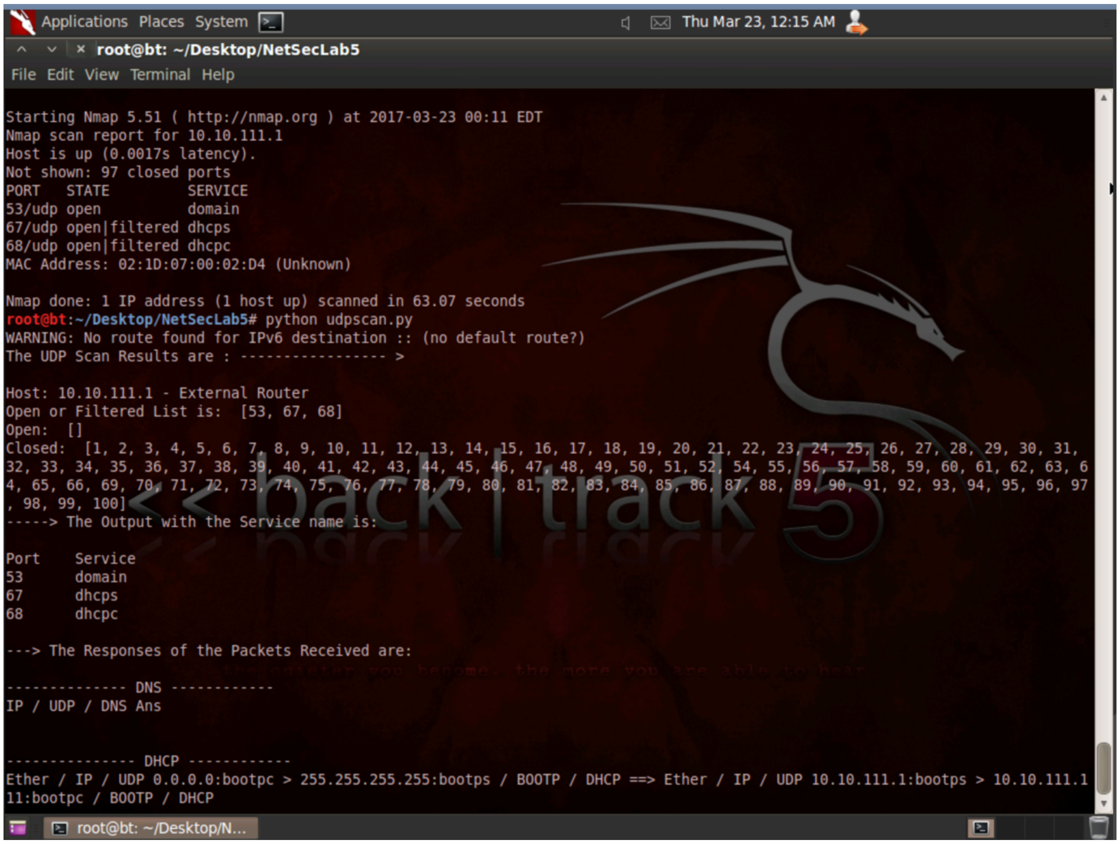
- Used Lists for categories Open, Closed and openORfiltered sets
- Hardcoded IP of the External Router – 10.10.111.1
- Increased timeout and retry to compensate the network congestion
- Iterated the Destination port of the UDP setting from 1 -> 101 for scanning the 100 ports
- A manual retry handle to retry for none received has been added

Code:

```
########################################################################################
###########
#
#                                 UDP Port Scanning Using Scapy
#
#    Author    : Srinivas Piskala Ganesh Babu
#    References: www.secdev.org/projects/scapy
#
#    Goal:
#         * Perform UDP Scan and Retrieve the Results to be Open, Closed or
Open|Filtered
#         * Perform Service Discovery for the respective ports
#         * Craft a Packets corresponding to the Service and Send to the Open Port
#
#   Solution:
#         * Performed UDP Scan
#         * Handled Network congestion by managing no response with retry argument and
manual handle
#         * Analyzed the Response to be:
#                   - None          - Open or Filtered
#                   - UDP Response  - Open
#                   - ICMP Response - Filtered
#                   - ICMP Response -
#
#        * Handled the Port:Service relationship with a dictionary
#        * Once the Service is discovered, the respective packet is crafted and sent
#                                     - This case DNS and DHCP Packets are handled
#
########################################################################################
#############

# Headers
```

```
import sys
from scapy.all import *

# Input Parameters

dst_ip = "10.10.111.1" # IP of the External Router

# Output Parameters - List to present the Output in various category sorted form

Open = []
Closed = []
openORfilter = []

# Iterating for Destination Ports 1->100 as mentioned in the quesiton
for i in range(1,101):
# Scapy UDP SYN Packet Builder - IP Layer and TCP Layer
 packet = IP(dst = dst_ip)/UDP(dport= i)
# Send/Receive handle - sr1 - Fetch Only Answer with increased timeout (5 Sec) and 
Verbose level 0 and Retry 2 times for Lost packet
 udp_scan = sr1(packet, verbose=0,retry=2, timeout=5)

# Retry Handle to retry manually one time
 if("None" in str(type(udp_scan))):
     udp_scan = sr1(packet, verbose=0, timeout=5)

# Sorting the Output to Filtered or Open or Closed List based ont he received packet
 if("None" in str(type(udp_scan))):
     openORfilter.append(i)
 elif(udp_scan.haslayer(UDP)):
     Open.append(i)
 elif(udp_scan.haslayer(ICMP)):
      if (udp_scan["ICMP"].type == 3 and udp_scan["ICMP"].code == 3):
          Closed.append(i)
      else:
          openORfilter.append(i)
 else:
   print "Unknown Received !"

# Output Print
print "The UDP Scan Results are : ----------------- >\n"
print "Host: 10.10.111.1 - External Router"
print "Open or Filtered List is: ", openORfilter
print "Open: ", Open
print "Closed: ",Closed

#--------------------> Service Name Discovery

# Service and Port name List - Obtained from IANA
service_list = {"53":"domain", "67":"dhcps","68":"dhcpc"}
# Response Handle to Record the response of Open Ports
response = []

# Packet Construction
# Packet Crafting for the Open Ports Service Name - 53 67 68
Packet53 = IP(dst= dst_ip)/UDP(dport=53)/DNS(qd=DNSQR(qname="www.google.com"))
# Based on secdev.org documentation - Scapy configuration for check ip to false for the 
DHCP to work
conf.checkIPaddr = False
# Retrieve the HW Mac of the Interface
fam,hw = get_if_raw_hwaddr(conf.iface)
Packet68 = 
Ether(dst="ff:ff:ff:ff:ff:ff")/IP(src="0.0.0.0",dst="255.255.255.255")/UDP(sport=68,dpor
t=67)/BOOTP(chaddr=hw)/DHCP(options=[("message-type","discover"),"end"])

# Check for Port Service Existence and Packet Send based on the respective Service
# Output for Port and Service Name
print "-----> The Output with the Service name is:"
print "\nPort\tService"
for port in openORfilter:
    if str(port) in service_list:
       print "%s\t%s" % (str(port),service_list[str(port)])

print "\n---> The Responses of the Packets Received are:\n "

for port in openORfilter:
       if str(port) == "53":
           print "--------------- DNS ------------"
           dns = sr1(Packet53,verbose=0,timeout=2)
           print dns.summary()
           print "\n"
       elif str(port) == "67":
           print "---------------- DHCP ------------"
           dhcp, unans = srp(Packet68,verbose=0,timeout=2)
           dhcp.summary()
```

Result Screen: Compared the results with the nmap and script, screen below


Applications Places System
Thu Mar 23, 12:15 AM
root@bt: ~/Desktop/NetSecLab5
File Edit View Terminal Help
root@bt: ~/Desktop/N...

```
Starting Nmap 5.51 ( http://nmap.org ) at 2017-03-23 00:11 EDT
Nmap scan report for 10.10.111.1
Host is up (0.0017s latency).
Not shown: 97 closed ports
PORT     STATE         SERVICE
53/udp   open          domain
67/udp   open|filtered dhcps
68/udp   open|filtered dhcpc
MAC Address: 02:1D:07:00:02:D4 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 63.07 seconds
root@bt:~/Desktop/NetSecLab5# python udpscan.py
WARNING: No route found for IPv6 destination :: (no default route?)
The UDP Scan Results are : ---------------- >

Host: 10.10.111.1 - External Router
Open or Filtered List is:  [53, 67, 68]
Open:  []
Closed:  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,
32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 6
4, 65, 66, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97
, 98, 99, 100]
-----> The Output with the Service name is:

Port    Service
53      domain
67      dhcps
68      dhcpc

---> The Responses of the Packets Received are:

-------------- DNS ------------
IP / UDP / DNS Ans


--------------- DHCP ------------
Ether / IP / UDP 0.0.0.0:bootpc > 255.255.255.255:bootps / BOOTP / DHCP ==> Ether / IP / UDP 10.10.111.1:bootps > 10.10.111.1
11:bootpc / BOOTP / DHCP
```

3. ***For an open UDP port found OPEN on rtr, lookup the service name associated with the port number and send a well-formed UDP packet using scapy for that service to the port to verify it is running a service and what that service is. Look up service names and transport protocol numbers from the number registry in iana.org. You only need to do this for ports that you found OPEN on rtr.***

Solution Highlights:

- ✓ Obtained the Service:Port Relationship results from the IANA website and stored in to dictionary for reference – **{“port” : “service”}**

- ✓ Ports which were open or open|filtered were cross checked with this dictionary for any known service existence
- ✓ If the service exists the respective is printed in the format and the packet with respect to the service is crafted and sent to the device, finally the response is caught and printed
- ✓ Compared with the NMAP Results and Confirmed it to match the findings
- ➔ **Results were: Open|Filtered: 53,67,68, Closed: [List of all ports from 1-100 except 53,67,68], Open: 0**

Program Flow:

- Performed a UDP Packet Send to port scan
- If there is no response, Resend Accomplished with Retry Argument as well as with manual handle
- Analyzed the results to be:
  - None - Open|Filtered
  - UDP Response - Open
  - ICMP Response
    - Type 3 and Code 3 – Destination and Port Unreachable – Closed
    - Else consider as - Open|Filtered

Highlights:

- Known port results from IANA is stored in a dictionary to refer the open|filtered ports – **{“port” : “service”}**
- If the Service is discovered then a packet with respect to service is crafted and sent
- In this case,

  Port 53: DNS – Created a DNS Query and Sent to the Service

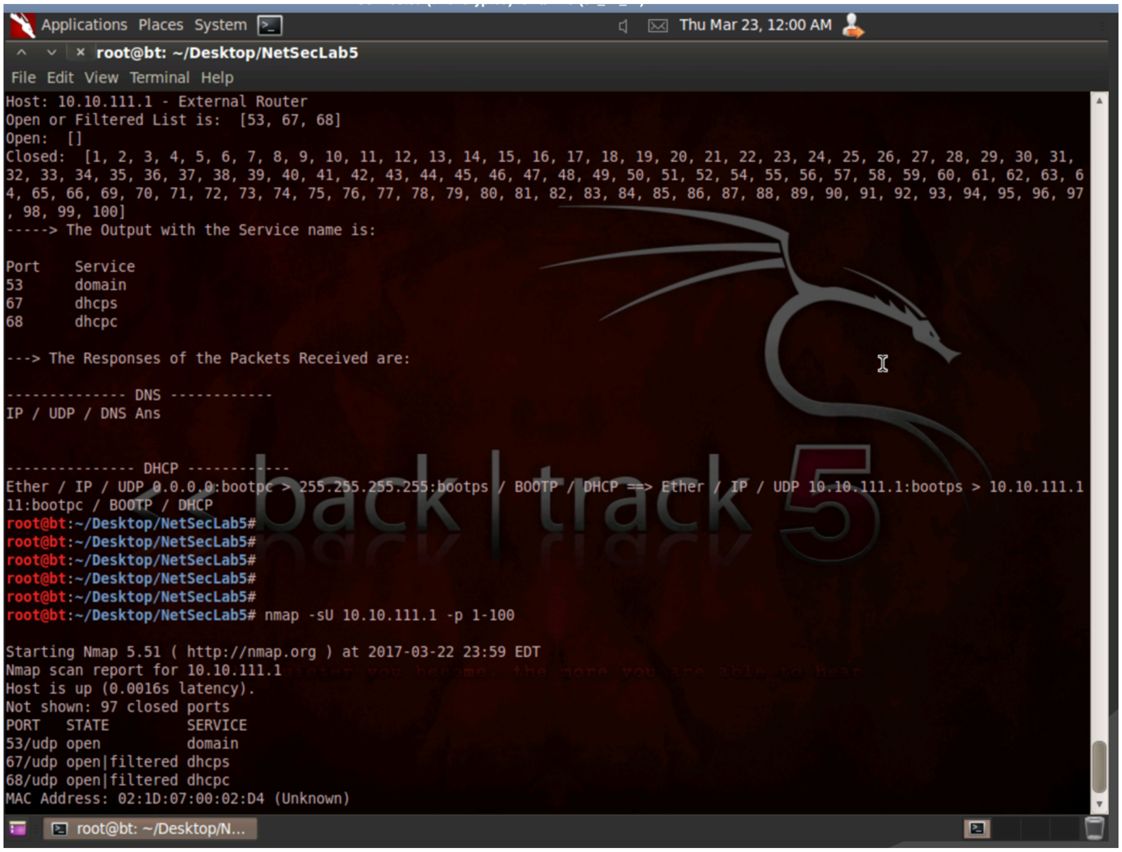
  Port 67,68: DHCP – Created a DHCP Discover Packet and Sent to the Service

Code Segment where the logic exists:

```
#--------------------> Service Name Discovery

# Service and Port name List - Obtained from IANA stored in a DICT
service_list = {"53":"domain", "67":"dhcps","68":"dhcpc"}
# Response Handle to Record the response of Open Ports
response = []

# Packet Construction
# Packet Crafting for the Open Ports Service Name - 53 67 68
Packet53 = IP(dst= dst_ip)/UDP(dport=53)/DNS(qd=DNSQR(qname="www.google.com"))
# Based on secdev.org documentation - Scapy configuration for check ip to false for the
DHCP to work
conf.checkIPaddr = False
# Retrieve the HW Mac of the Interface
fam,hw = get_if_raw_hwaddr(conf.iface)
Packet68 =
Ether(dst="ff:ff:ff:ff:ff:ff")/IP(src="0.0.0.0",dst="255.255.255.255")/UDP(sport=68,dpor
t=67)/BOOTP(chaddr=hw)/DHCP(options=[("message-type","discover"),"end"])

# Check for Port Service Existence and Packet Send based on the respective Service
# Output for Port and Service Name
print "-----> The Output with the Service name is:"
print "\nPort\tService"
for port in openORfilter:
    if str(port) in service_list:
       print "%s\t%s" % (str(port),service_list[str(port)])

print "\n---> The Responses of the Packets Received are:\n "

for port in openORfilter:
       if str(port) == "53":
           print "-------------- DNS ------------"
           dns = sr1(Packet53,verbose=0,timeout=2)
           print dns.summary()
           print "\n"
       elif str(port) == "67":
           print "--------------- DHCP ------------"
           dhcp, unans = srp(Packet68,verbose=0,timeout=2)
           dhcp.summary()
```

Output Screen:

- DNS - As there is no root server connections the response is plain with the fields
- DHCP – The response of dhcp discover is fetched back


Applications Places System
Thu Mar 23, 12:00 AM
root@bt: ~/Desktop/NetSecLab5
File Edit View Terminal Help
back | track 5
root@bt: ~/Desktop/N...

```
Host: 10.10.111.1 - External Router
Open or Filtered List is:  [53, 67, 68]
Open:  []
Closed:  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,
32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 6
4, 65, 66, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97
, 98, 99, 100]
-----> The Output with the Service name is:

Port    Service
53      domain
67      dhcps
68      dhcpc

---> The Responses of the Packets Received are:

-------------- DNS ------------
IP / UDP / DNS Ans


--------------- DHCP ------------
Ether / IP / UDP 0.0.0.0:bootpc > 255.255.255.255:bootps / BOOTP / DHCP ==> Ether / IP / UDP 10.10.111.1:bootps > 10.10.111.1
11:bootpc / BOOTP / DHCP
root@bt:~/Desktop/NetSecLab5#
root@bt:~/Desktop/NetSecLab5#
root@bt:~/Desktop/NetSecLab5#
root@bt:~/Desktop/NetSecLab5#
root@bt:~/Desktop/NetSecLab5#
root@bt:~/Desktop/NetSecLab5# nmap -sU 10.10.111.1 -p 1-100

Starting Nmap 5.51 ( http://nmap.org ) at 2017-03-22 23:59 EDT
Nmap scan report for 10.10.111.1
Host is up (0.0016s latency).
Not shown: 97 closed ports
PORT   STATE         SERVICE
53/udp open          domain
67/udp open|filtered dhcps
68/udp open|filtered dhcpc
MAC Address: 02:1D:07:00:02:D4 (Unknown)
```